

Fondamenti di Informatica 1 - anno accademico 200? - 200?
Prova di Programmazione, appello simulato

Si vuole fornire una parziale realizzazione di una agenda elettronica, che consenta di memorizzare un insieme di impegni. Gli impegni memorizzati nell'agenda saranno coppie di tipo "chiave valore", dove il campo chiave è un numero in formato `int` che specifica il livello di priorità di un impegno mentre il campo valore è una stringa contenente un promemoria dell'impegno. Si assume di utilizzare quattro livelli di priorità, da 0 a 3 estremi inclusi, dove il livello 0 significa "massima priorità" ed il livello 3 significa "minima priorità".

Il candidato ha a disposizione il file **AgendaTester.java**, il quale contiene classi e interfacce da utilizzare e/o completare secondo quanto qui indicato:

- **PriorityQueue**. Interfaccia che definisce il tipo di dati astratto "coda di priorità". *Leggere con attenzione* i commenti all'interfaccia ed ai suoi metodi. *Non modificare*.
- **EmptyQueueException**. Eccezione che segnala che la coda di priorità è vuota. *Non modificare*.
- **Agenda**. Classe che implementa l'interfaccia **PriorityQueue**. *Completare la classe* scrivendone la parte privata e realizzandone i metodi pubblici. L'agenda conterrà coppie di tipo "chiave valore" appartenenti alla classe **Impegno**, realizzata come classe interna di **Agenda** e il cui codice *non va modificato*. Si richiede inoltre di *realizzare un metodo toString per la classe Agenda*, che restituisca una stringa contenente gli elementi secondo il seguente formato: (1) ogni coppia viene scritta su una riga diversa, e (2) all'interno di ogni riga la coppia viene scritta seguendo il formato specificato dal metodo `toString` della classe **Impegno** (si veda il corpo di tale metodo).
- **AgendaTester**. Classe che collauda l'agenda. *Completare la classe* in modo che il suo metodo `main` realizzi il seguente comportamento:

- Crea un oggetto di tipo **Agenda**, vuoto.
- Accetta ripetutamente comandi dall'utente, introdotti da tastiera, finché l'utente non introduce il comando di terminazione del programma. I comandi disponibili sono:

Comando	Significato
"I"	Inserisce un impegno. In questo caso il programma chiede di inserire da tastiera una riga di testo nel formato "chiave valore", ed inserisce l'impegno nell'agenda (ad es. la riga "0 Studiare polimorfismo!" inserisce un impegno con priorità 0 e promemoria "Studiare polimorfismo!").
"R"	Rimuove il primo impegno di priorità massima (ovvero chiave=0). In questo caso il programma rimuove l'impegno dall'agenda e ne stampa il promemoria a standard output.
"L"	Legge il primo impegno di priorità massima. In questo caso il programma stampa il promemoria dell'impegno a standard output (senza rimuovere l'impegno dall'agenda).
"Q"	Termina (Quit) il programma.

- Dopo ognuna delle operazioni sopra elencate stampa il contenuto aggiornato dell'agenda.

Nello svolgimento si possono usare, se ritenute utili, solo le seguenti classi della libreria standard:

- tutte le classi dei package `java.lang` e `java.io`;
- le classi `Scanner`, `StringTokenizer` e `NoSuchElementException` del package `java.util`.

Usando il file **impegni.txt** allegato, il programma può essere provato con il comando

java AgendaTester < impegni.txt

Al termine della prova lasciare nella directory di lavoro tutti i file. La prima riga del file .java dovrà contenere un commento con nome/cognome del candidato, matricola, data, numero postazione. Prima di uscire dall'aula consegnare questa scheda compilata.

Cognome e Nome:	Corso di Laurea:
Matricola:	Postazione: ADT
Consegno l'elaborato.	Non consegno l'elaborato e mi ritiro dall'esame.
Firma	Firma